Cryptography -
A Crash Overview

Stanisław Radziszowski
Rochester Institute of Technology
spr@cs.rit.edu

Gdańsk, November 2010
Rochester, January 2015
Cryptography

goals

Desired security properties in the digital world:

- confidentiality, secrecy
- data integrity
- authentication, of data origin and entity
- non-repudiation
Cryptography
limits

Cryptography is an important, but only a relatively small part of security:

- right choice of tools is hard
- implementation errors are common
- variety of side-channel attacks can bypass best crypto
- social attacks
Cryptography

basic scenario

[Stinson]
Primitives, algorithms and protocols can be **unkeyed**, **symmetric-key** or **public-key**

**Unkeyed**
- hashing, SHA-family (large part of these lectures)
- one-way permutations exist, or \( \textbf{NP} \) is not that much ...

**Use**
- hash and sign
- random sequences - Blum-Blum-Shub BBS generator, stream cipher outputs, \( H(n), H(n + 1), H(n + 2), \ldots \)
- many other ...
Hash in Use
message authentication - clear and encrypted

[Stallings]
SHA-3 = Keccak 2007 – 2050
the main hash you will use

- Winner of the 2007-2012+ NIST SHA-3 competition
  Draft FIPS 202, May 2014
- Team (from STMicroelectronics and NXP Semiconductors):
  Guido Bertoni, Joan Daemen (of AES fame),
  Michaël Peeters, Gilles Van Assche
- hashing SHA3-224, SHA3-256, SHA3-384, SHA3-512
- extendable-output function SHAKE128, SHAKE256
- Elegant, convincing sponge design, ideas from *Grindahl*
- Runs on a $5 \times 5 \times 2^l$ cube of bits,
  recommended 1600-bit state ($l = 6$)
SHA-3/SHAKE/Keccak sponge

- $r$ absorption rate
- $c$ security capacity
- $f$ crypto reshuffling permutation
- $p$ absorbed message
- $z$ squeezed output
Cryptography
shared-key primitives and algorithms

Symmetric keys

- block ciphers since 1970’s
  IBM’s Lucifer,
  DES - Data Encryption Standard, 3DES
  IDEA - International Data Encryption Algorithm
  AES - Advanced Encryption Standard
- stream ciphers, RC4 - also can come from counter mode of block ciphers or hash functions
- MAC, HMAC - message authentication codes
- PRNG - pseudo-random number generators

Feistel cipher

[Wikipedia]
all steps invertible in Galois fields

- Rijndael by Vincent Rijmen and Joan Daemen, BE
- winner of the NIST cipher 1997 – 2001 competition
- state is a $4 \times 4$ matrix of bytes in $GF(2^8)$

[Wikipedia]
Block Cipher in Use

shared key

Counter (CTR) mode encryption

[Wikipedia]
Cryptography
public-key primitives and algorithms

Public keys:

- Public-key cryptosystems
  - RSA - Rivest, Shamir, Adleman
  - ElGamal, McEliece cryptosystems
  - ECC - elliptic curve cryptosystems

- Signatures
  - DSS/DSA - Digital Signature Standard/Algorithm
  - ECDSA - Elliptic Curve Digital Signature Algorithm

- PKI - public-key infrastructure, only if we had it right :-(
  - DH - Diffie-Hellman key agreement
    - key management, distribution and X.509

- Homomorphic cryptography - Paillier, Gentry
Main Public-Key Systems in Use
RSA and ECC

RSA by Rivest-Shamir-Adleman, 1977 has an edge over ECC, because
- it is simple and well understood
- links nicely to basic number theory
- deployed earlier on many systems

ECC by Koblitz-Miller, 1985 has an edge over RSA, because
- it uses short keys (163+ bits ECC vs. 1024+ bits RSA)
- delivers much better performance
- ECC uses great theory of elliptic curves on top of classical number theory used by RSA

Prediction: finally ECC will take over
Public-key System in Use

signature by hash and public-key encryption

- **Signing**
  - Data
  - Hash function
  - Hash: 101100110101
  - Encrypt hash using signer's private key
  - Signature: 111101101110
  - Attach to data
  - Digitally signed data

- **Verification**
  - Digitally signed data
  - Decrypt using signer's public key
  - Signature: 111101101110
  - Hash function
  - Hash: 101100110101
  - ?
  - Hash: 101100110101

If the hashes are equal, the signature is valid.

[Wikipedia]
Other composite and special functionalities

- Zero-knowledge protocols
- Authenticated encryption
  CAESAR competition: 2012 – 2017/2018
- Electronic cash: untraceable, no double-spending, bank-shop-customer roles, central bank
- Cryptocurrencies, SHA or scrypt POW based: Bitcoin, fully distributed, anonymity questioned ...
  Zerocoin, Litecoin, Darkcoin, Mixcoin, more coming ...
- Electronic voting: no individual vote audit
- Oblivious transfer, two millionaires problem
- Quantum and post-quantum cryptography
  quantum key distribution (Y), quantum computing (N)
Mathematics in Cryptography

Math in primitives

- Keyless: so far mostly bit juggling, we will see soon what kind of math is in SHA-3
- Shared-key: much more since AES ’2001, mostly around binary Galois fields $GF(2^k)$
- Public-key: heavy use of number theory, now essentially in all PKC, including ECC

Math in cryptanalysis

- Linear and differential cryptanalysis
- Probability and statistics, random oracle models
- Number theoretical algorithms: primality, factoring
- Discrete logarithms: cyclic group discovery, index calculus, counting points on elliptic curves, theory of elliptic curves
Security engineer must consider:

- **Level of security.** Or, how many security bits you need.
- **Functionality.** Or, how primitive are the primitives.
- **Performance.** Or, how fast is fast enough.
- **Simplicity.** Is there still anybody who can understand it?

Each party stresses a different measure:

- **risk** (politicians)
- **cost** (managers)
- **use** (most of us)

Can security/software engineer satisfy all of them?
References

  
  http://www.cacr.math.uwaterloo.ca/hac

